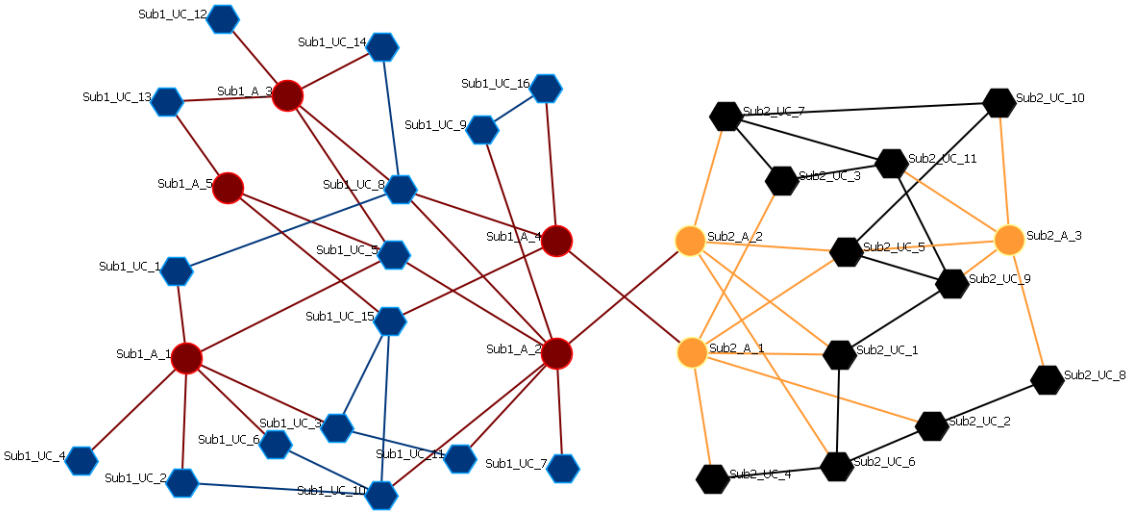
17. Towards the Use of Network Analysis Method In Analysing Node Properties In A System Model

Li Jiang and Hossein Seif Zadeh
Joint Operation Division, DSTO

Abstract

Model-based system engineering methodologies advocate using system models as the main vehicle in system engineering processes[12]. In this methodology, a system model represents the relationships and interaction between the entities being modelled. **Figure 2** depicts an example of such abstraction of the interaction within and between two subsystems.



(a) The first component network **(b) The second component network**

Legend: Filled circles represent actors or agents
Filled diamonds represents use cases or components
Different colours are used to distinguish actors (agents) or use cases (components) in each subsystem.

Figure 2 A sample component network of two subsystems

As a result of the difficulty in understanding complex relationships within comprehensive systems models, there is a need for a systematic approach in assessing properties of such models[13].

[12] Estefan, J. (2008). *Survey of model-based systems engineering (MBSE) methodologies*. Pasadena, California. USA, Jet Propulsion Laboratory, California Institute of Technology

[13] Brooks, R. J. and A. M. Tobias (1996). *Choosing the Best Model: Level of Detail, Complexity, & Model Performance*, Mathematical and Computer Modelling, Volume 24, Number 4, August 1996 , pp1-14 testing

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
FEB 2013	**N/A**	**-**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Towards the Use of Network Analysis Method In Analysing Node Properties In A System Model	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Joint Operation Division, DSTO**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
See also ADA585222. Proceedings of the 2012 Model-Based Systems Engineering Symposium, 27 - 28 November 2012, DSTO Edinburgh, South Australia., The original document contains color images.

14. ABSTRACT
Model-based system engineering methodologies advocate using system models as the main vehicle in system engineering processes12. In this methodology, a system model represents the relationships and interaction between the entities being modelled. Figure 2 depicts an example of such abstraction of the interaction within and between two subsystems.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**SAR**	**16**	

Lacking evaluation mechanism for system models presents three major problems:

(1) difficulty in understanding fundamental properties of the model which are often attributed as a major reason for failure of the system;

(2) lack of a systematic and efficient mechanism in ensuring consistency of the model through all stages of process, system, and product development[14]; and

(3) difficulty in understanding which components perform critical functions, and which components serve as a bridge between sub-systems.

This paper presents a two-step approach in assessing properties and consistency of the model. The definitions of the properties and consistency are briefly discussed below:

- Properties are defined based on a set of network science measures[15]. To use the network science measures, the relationships between entities in the system model are represented as an entity network (see Figure 1 for a simple example). The network measures can be computed and the results of the computation can be explained meaningfully within the system engineering discipline.
- Consistency refers to the congruent between entities or artefacts developed in the system development process. These measures can be quantitative or qualitative.

Jiang et al[16] have shown that, in the context of software development, analysing properties of a model provides meaningful feedback for the purpose of design and system verification processes.

The proposed approach provides a practical mechanism for analysing properties of the system. The major contribution of this work is two folds:

(1) properties of system models can be used at both network and node level, containing critical information on the overall entity network, and

(2) consistency-assessment measures provide a mechanism to verify consistency of the system model.

The implication and significance of using properties of nodes within the context of system engineering are also discussed.

[14] Van Der Straeten, R., T. Mens, et al. (2003). *Using description logic to maintain consistency between UML models*. «UML» 2003-The Unified Modeling Language. Modeling Languages and Applications: 326-340.
[15] Wasserman, S. and K. Faust (1995). *Social Network Analysis: Methods and Applications*. Cambridge, University of Cambridge Press
[16] Jiang, L., K. M. Carley, et al. (2012). *The Impact of Component Interconnections On Software Quality: A Network Analysis Approach*. The 2012 IEEE International Conference on Systems, Man, and Cybernetics (IEEE SMC 2012)

Presenter Biographies

Dr. Li Jiang obtained his PhD in Nov. 2005. He has more than 50 publications with more than 30 published in the reputed international journals and conferences. He won many awards including Canadian PhD scholarship in 2002, Canadian visiting fellowship from Natural Sciences and Engineering Research Council of Canada in 2006. After completion of his PhD., Dr. Li Jiang started to work as a lecturer in the Department of Computer Science at the University of New Brunswick, Canada, in 2005 and a lecturer in the School of Computer Science at the University of Adelaide, Australia since Nov. 2006. Dr. Jiang started to work at DSTO in Canberra from August, 2012. Dr. Jiang has been a visiting scientist at the University of Carnegie Mellon University, University of Calgary, and University of Nottingham in 2011, 2001 and 1995 respectively.

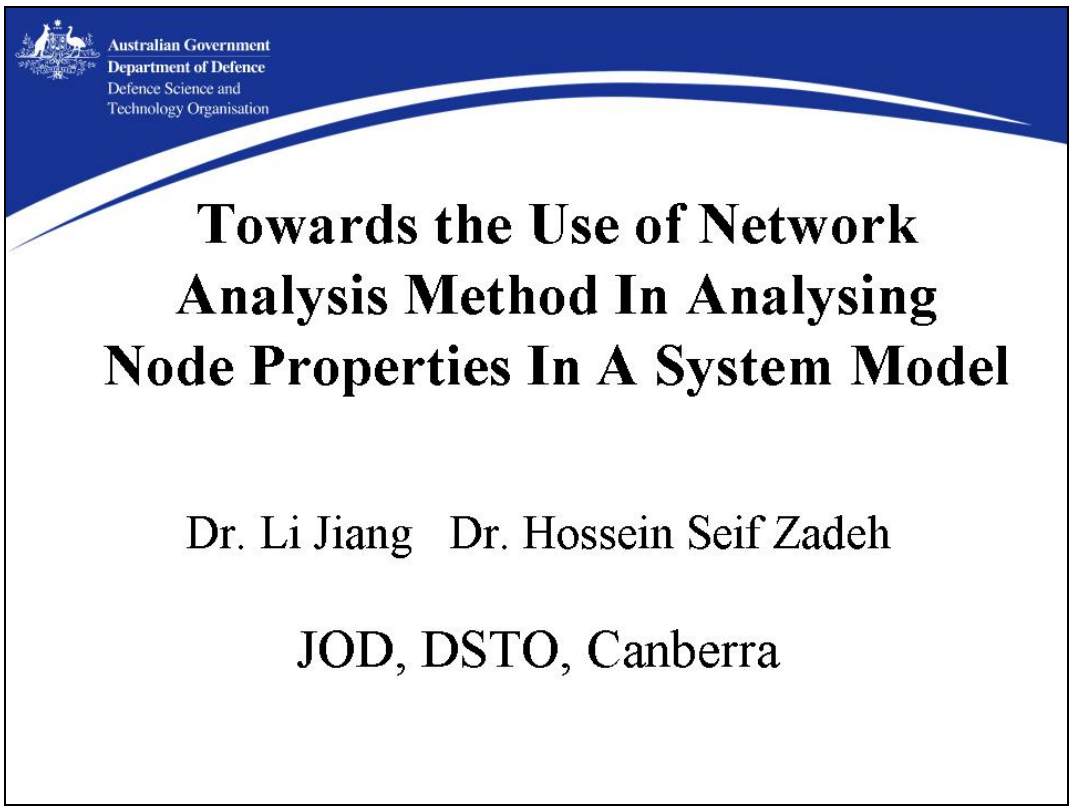
Besides having academic experience in Canada and Australia, Dr. Jiang also has more than 7 years working experiences in software industry both in China and Canada as programmer, analyst, architect, and project manager.

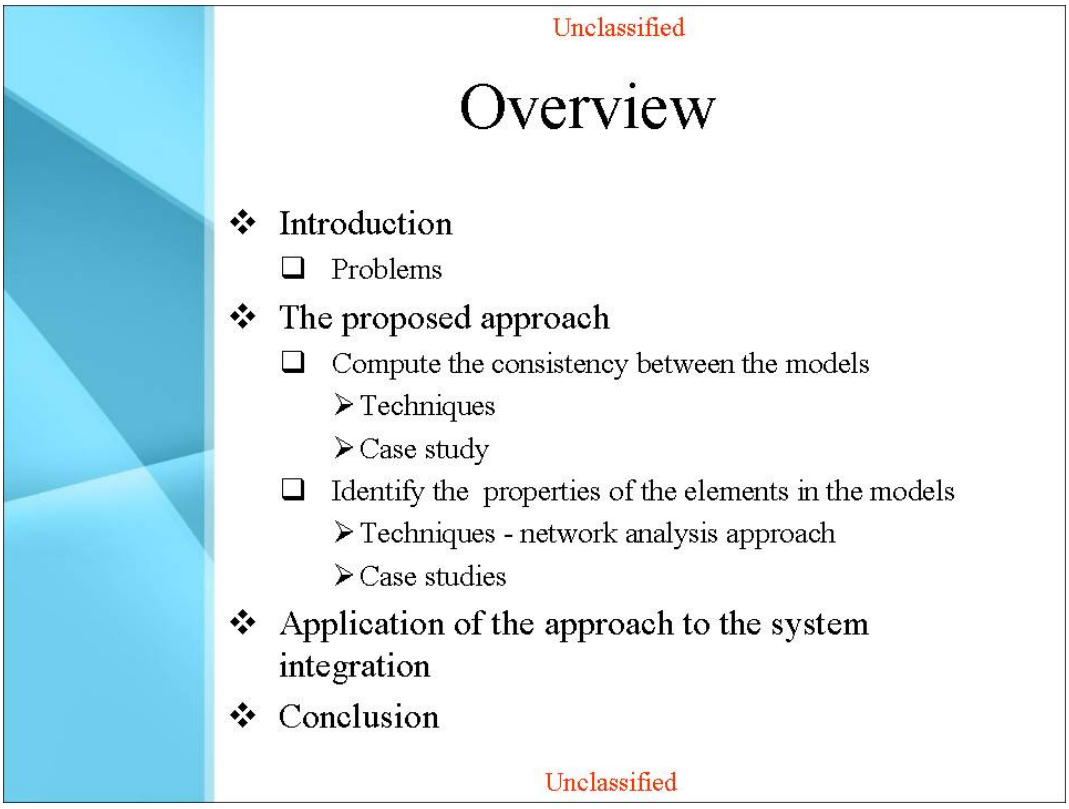
Hossein Seif Zadeh's career includes positions as research scientist, senior IT manager, senior project manager, management consulting, system analyst, and educator. Dr. Seif Zadeh's experience combines disciplines of mechanical and aerospace engineering in the one hand and management and information systems in the other. He has researched and published in fields as diverse as manoeuvre control of satellites to the innovative applications of information systems in healthcare.

After a management experience looking after a large-scale IT department with 15,000+ clients, and a successful academic career, Hossein now holds the position of Science Team Leader at DSTO Fairbairn. His research, linking the diverse fields of engineering, management, and IT has attracted over $1,000,000 in grants, awards, scholarships and contracts, from organizations such as the Australian Research Council and Department of Defence. In 2004, Hossein was a visiting scholar at Linkoping University, Sweden, and in 2009/2010, was a Distinguished Visiting Scholar at IBM Almaden Research Labs, San Jose, USA.

Hossein is a continuing reviewer of multiple international journals and conferences, was associate editor of IT Services track in ICIS 2010 and is mini-track chair of AmCIS 2011, and has been a session organizer and reviewer of IEEE Aerospace conference since 2002. In 2010, Hossein was a recipient of the prestigious and internationally-recognized IBM Faculty Award, and in 2012 was selected as a Fellow of Schoeller Research Center in Germany.

Presentation





Unclassified

Introduction

Compute the consistency

Identify the properties

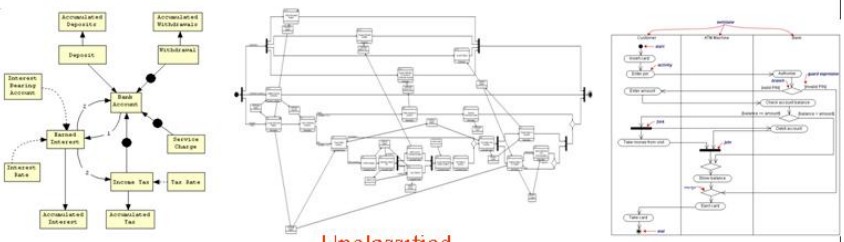
Application to the system integration

Conclusion

Introduction

Model-based system engineering (MBSE)

- MBSE is the formalized application of modelling to support system requirements, design, analysis, verification and validation activities in the system engineering life cycle.
  - Requirements Models – Requirement Diagram
  - Design Models - Package Diagram, Sequence Diagram, Activity Diagram, State Machine Diagram, etc.



Unclassified

Unclassified

Introduction

Compute the consistency

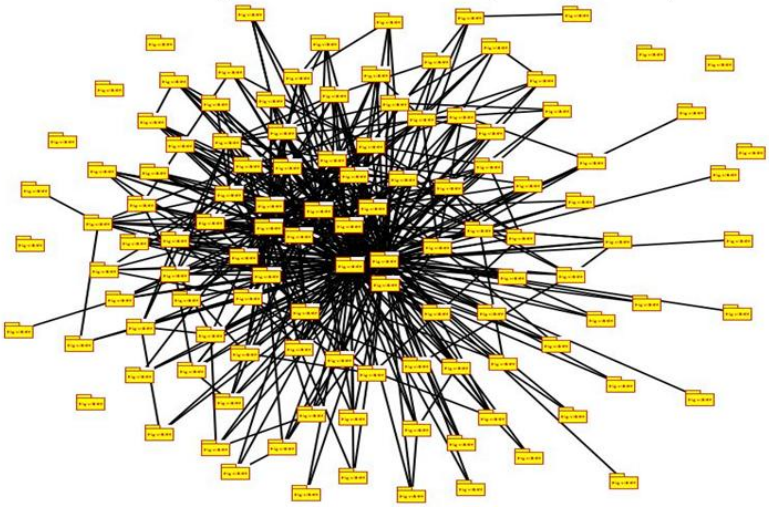
Identify the properties

Application to the system integration

Conclusion

Introduction (Cont'd)

- Problems
  - Hard to ensure the consistency between the designs and evolutions of design.
  - Hard to identify the critical elements (nodes) in the system



Unclassified

Introduction
Compute the consistency
Identify the properties
Application to the system integration
Conclusion

Introduction (Cont'd)

- How to verify and evaluate the system models remains challenges in both industry and academia
  - Status quo in industry practices
  - Research – major focus on mathematical approaches
    - model-checking and automated theorem proving
    - using description logic to maintain consistency
- Lacking evaluation and/or verification mechanism for system models presents three major problems
  1) lack of a systematic and efficient mechanism in ensuring consistency of the model
  2) difficulty in understanding which components perform critical functions
  3) difficulty in understanding fundamental properties of the model

Unclassified

Unclassified

Introduction
The Proposed Approach
Compute the consistency
Identify the properties
Application to the system integration
Conclusion

The Proposed Approach

- An approach is proposed for verification and evaluation of the models.

The approach include two parts:

(1) Define a set of measures to compute the consistency between the models.

(2) Using several network measures to identify the properties of the elements in the model.
  - Compute the complexity of the model
  - Compute the properties of the elements in the model.

Unclassified

Unclassified

Introduction
The Proposed Approach
Compute the consistency
Identify the properties
Application to the system integration
Conclusion

The Proposed Approach (Cont'd)

- Assumption with the approach:
  - The system design following the system engineering process and SysML (or UML) are used in the design.
  - The relationships between the requirements, objects, components or package of the system in the system models are well-established.
  - Targeting on the project covering the entire system development lifecycle.
- The applicability of the approach to the system integration is briefly discussed at the end of the presentation.

Unclassified

Unclassified

Introduction
The Proposed Approach
Compute the consistency
Identify the properties
Application to the system integration
Conclusion

Part 1: Compute The Consistency Between The Models

Step 1: Define a set of measures

- The measures are divided into following classes
  - Quantity metrics
    - counts of the design entities and relationships.
  - Complexity metrics
    - measure the relations between design entities and the structure of the proposed system architecture.
  - Quality metrics
    - measure the relationship between the desired and the actual characteristics of the architecture.

Unclassified

Unclassified

Introduction

The Proposed Approach

Compute the consistency

Identify the properties

Application to the system integration

Conclusion

Part 1: Compute The Consistency Between The Models (Cont'd)

- Examples of the proposed quantity metrics for evaluation of the models
  - Number of Diagrams
    - Package Diagrams, Use Case Diagrams, Sequence Diagrams, State Diagrams, Activity Diagrams, Requirements Diagram, Class Diagram
  - Number of entities
    - Requirements, Use Cases, Actors, Activities, Package
  - Number of design relationship type
    - Links between entities, Interactions, Activity Flows, State Transitions.

Unclassified

Unclassified

Introduction

The Proposed Approach

Compute the consistency

Identify the properties

Application to the system integration

Conclusion

Part 1: Compute The Consistency Between The Models (Cont'd)

- Examples of the proposed complexity metrics

$$OverallDesignComplexity = 1 - \left[\frac{No\_DesignEntities}{No\_Relationships + No\_Actors} \right]$$

$$UseCase\_Complexity = 1 - \left[\frac{No\_UseCase}{No\_Relationships + No\_Actors} \right]$$

$$Object\_Interation\_Complexity = 1 - \left[\frac{\frac{No\_of\_Object}{No\_of\_Object\_Interaction} + \frac{No\_of\_Classes}{No\_of\_Class\_Association}}{2} \right]$$

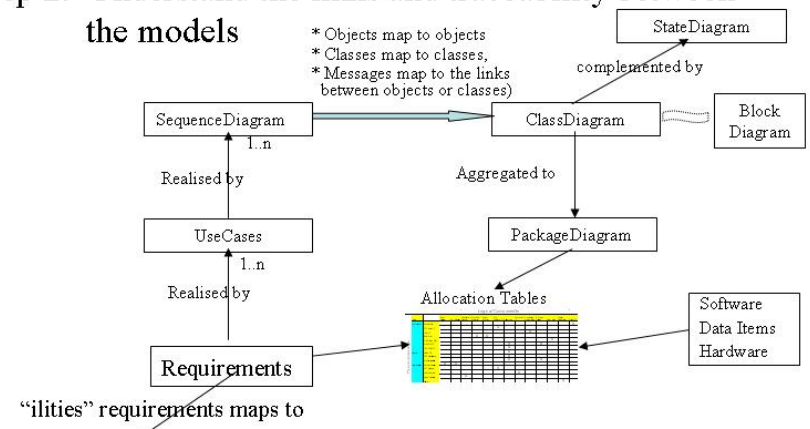
Unclassified

Unclassified

Introduction
The Proposed Approach
Compute the consistency
Identify the properties
Application to the system integration
Conclusion

Part 1: Compute The Consistency Between The Models (Cont'd)

Step 2: Understand the links and traceability between the models



"ilities" requirements maps to

Other System Design Consideration
- System architecture
- Hardware (such as safety and reliability)
- Network (such as, cable, hub, structure, security)
- Operation system and software

Unclassified

Unclassified

Introduction
The Proposed Approach
Compute the consistency
Identify the properties
Application to the system integration
Conclusion

Part 1: Compute The Consistency Between The Models (Cont'd)

Step 3: Define a set of consistency measures

$$DegreeOfConsistency_{req\_usecase} = \frac{No\_RequiremetnsRealisedByUseCases}{No\_Requirements}$$

$$DegreeOfConsistency_{usecase\_SeqD} = \frac{No\_UseCasesModelledBySequenceDiagram}{No\_UseCases}$$

$$DegreeOfConsistency_{SeqD\_ClassDiagram} = \frac{No\_ClassesInClassDiagram + No\_ObjectsInObjectDiagram}{No\_ClassesInSequenceDiagram + No\_ObjectsInSequenceDiagram}$$

$$DegreeOfConsistency_{classes\_methods} = 1 - \frac{No\_UndefinedMethodsReferences + No\_UnderfinedParameterReferences}{No\_DefinedMethods + No\_DefinedParameters}$$

Unclassified

Unclassified

Introduction
The Proposed Approach
Compute the consistency
Identify the properties
Application to the system integration
Conclusion

Part 1: Compute The Consistency Between The Models (Cont'd)

Case Study 1: Compute the consistency between the models

- Data Sources: Student Group's Design Documents:
- Information about the students:
  - Year 3 students from computer science, math and other engineering program.
  - Students are involved in Group Project with 5 to 6 group members.
  - Intensive one term-long project supervised by lecturers
  - The project is about developing a robot that can detect mines in the "battle field"
  - Students are guided through the entire engineering process from requirements gathering to the final deliverables
  - Students uses various engineering process models
  - SRS, SDD, SPMP are compulsory deliverables and presented during the processes of the project

Unclassified

Unclassified

Introduction
The Proposed Approach
Compute the consistency
Identify the properties
Application to the system integration
Conclusion

Part 1: Compute The Consistency Between The Models (Cont'd)

Case Study 1: Compute the consistency between the models

- Data Sources: Student Group's Design Documents:
- Information about the students:
- Results

	Group 1 (2010)	Group 15 (2010)	Group 5 (2011)	Group 4 (2011)
DegreeOfConsistency (Requirements and Usecases)	0.72	0.83	0.88	0.69
DegreeOfConsistency (Usecases and SequenceDiagram)	0.60	0.58	0.83	0.80
DegreeOfConsistency (SequenceDiagram and ClassDigram)	0.93	0.57	0.79	0.49
Overall Consistency	0.40	0.27	0.58	0.27
Average	0.78	0.69	0.84	0.67

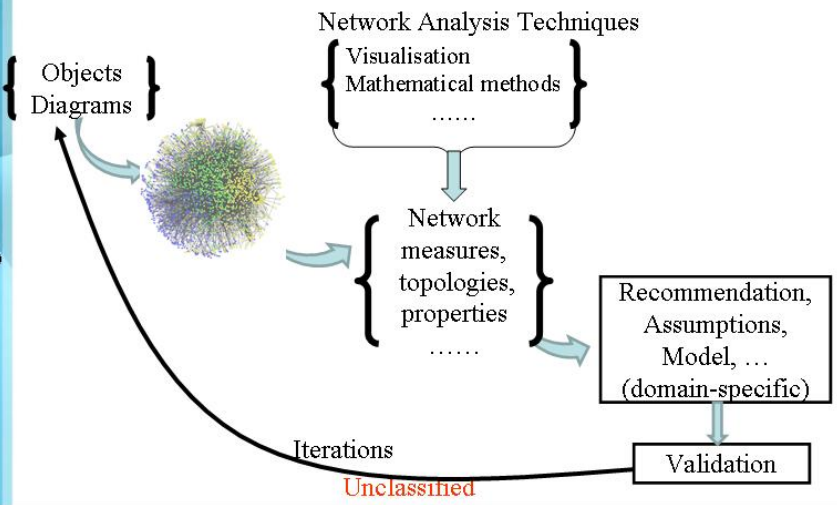
Unclassified

Unclassified

Introduction
The Proposed Approach
Compute the consistency
Identify the properties
Application to the system integration
Conclusion

Part 2: Identify the properties of the elements in the models

- **Using network analysis approach**



Unclassified

Unclassified

Introduction
The Proposed Approach
Compute the consistency
Identify the properties
Application to the system integration
Conclusion

Part 2: Identify the properties of the elements in the models (Cont'd)

- Examples of network measures used
  - Network level
    - Network Size, Link count, Density, Isolate count (Component count), Clustering coefficient
  - Node level
    - Degree centrality, Betweenness centrality, Eigenvector centrality, Closeness centrality
- Network analysis techniques used
  - Visualisation
  - Computation analysis
  - Statistical analysis

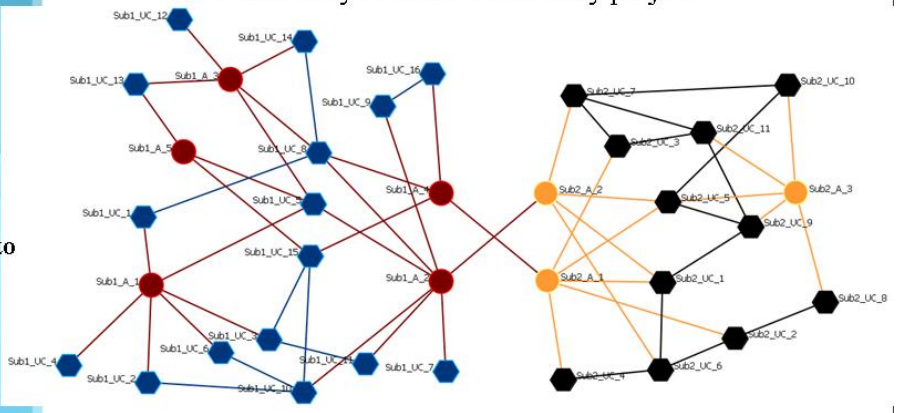
Unclassified

Unclassified

Part 2: Identify the properties of the elements in the models (Cont'd)

Introduction
The Proposed Approach
Compute the consistency
Identify the properties
Application to the system integration
Conclusion

- Case study 2
  - Analyse the critical nodes in the two use case diagrams for two subsystems for an industry project.



Note: Filled-circles represent actors or agents and filled-diamonds represents use cases

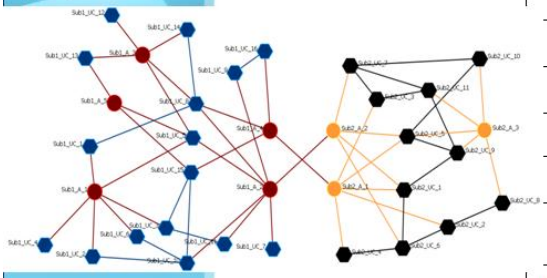
Unclassified

Unclassified

Part 2: Identify the properties of the elements in the model (Cont'd)

Introduction
The Proposed Approach
Compute the consistency
Identify the properties

- Case study 2
  - Analyse the critical nodes in the two use case diagrams for two subsystems for an industry project.
  - Compute the measures
  - Analysis the results



Rank	Total degree centrality	Betweenness centrality	Closeness centrality	Eigenvector centrality
1	Sub1_A_2	Sub2_A_1	Sub1_A_2	Sub2_UC_5
2	Sub1_A_1	Sub2_A_2	Sub1_A_4	Sub2_A_2
3	Sub2_A_1	Sub2_UC_5	Sub1_A_1	Sub2_A_1
4	Sub1_A_3	Sub1_UC_8	Sub2_A_1	Sub2_A_3
5	Sub1_UC_8	Sub2_UC_3	Sub2_A_2	Sub2_UC_9
6	Sub2_A_2	Sub2_UC_7	Sub1_A_3	Sub2_UC_1

Unclassified

Unclassified

Introduction

The Proposed Approach

Compute the consistency

Identify the properties

Application to the system integration

Conclusion

Part 2: Identify the properties of the elements in the model (Cont'd)

- The empirical verification of the model:
  - Project actual results obtained from programmers and testing engineers
    - Following use cases took more time to implement than other nodes:

      Sub1_A_2, Sub1_A_1, Sub2_A_1 , Sub1_A_3
    - Following use cases took more time to implement than other nodes and more test cases were required and implemented in the testing process than other nodes:

      Sub2_UC_5 , Sub1_UC_8, Sub2_UC_3, Sub2_UC_7

Unclassified

Unclassified

Introduction

The Proposed Approach

Compute the consistency

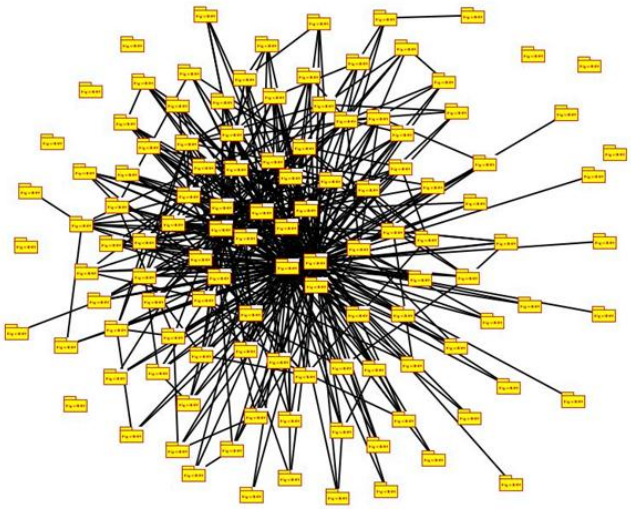
Identify the properties

Application to the system integration

Conclusion

Part 2: Identify the properties of the elements in the model (Cont'd)

- Case study 3
  - Visualisation of the system architecture



Unclassified

Introduction

The Proposed Approach

Compute the consistency

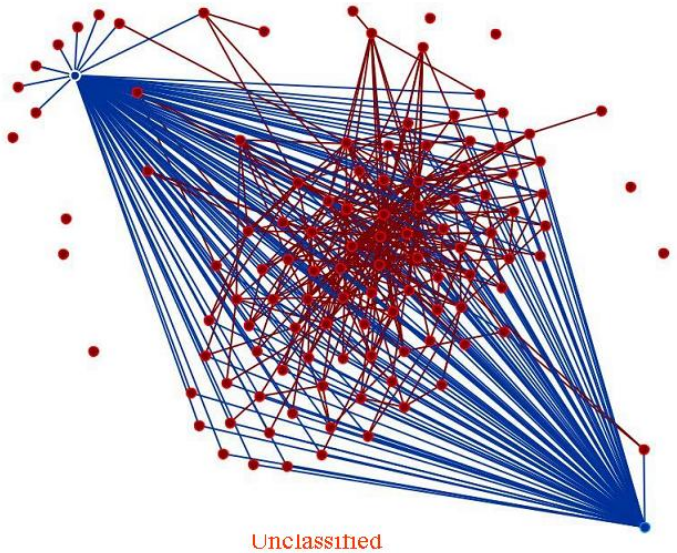
Identify the properties

Application to the system integration

Conclusion

Part 2: Identify the properties of the elements in the model (Cont'd)

Analysis Via Visualisation (2)- Release 1



Unclassified

Unclassified

Introduction

The Proposed Approach

Compute the consistency

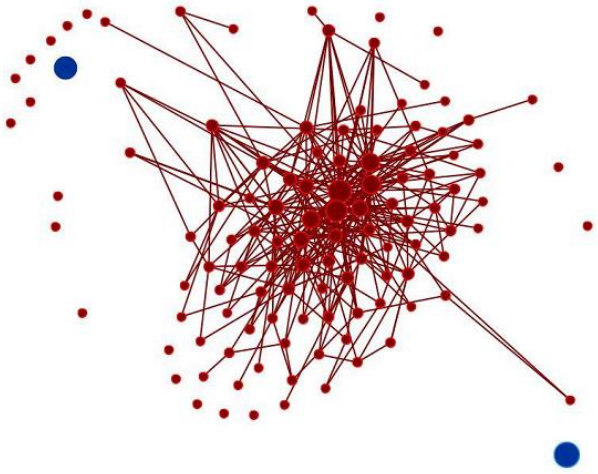
Identify the properties

Application to the system integration

Conclusion

Part 2: Identify the properties of the elements in the model (Cont'd)

Analysis Via Visualisation (3) – Release 1



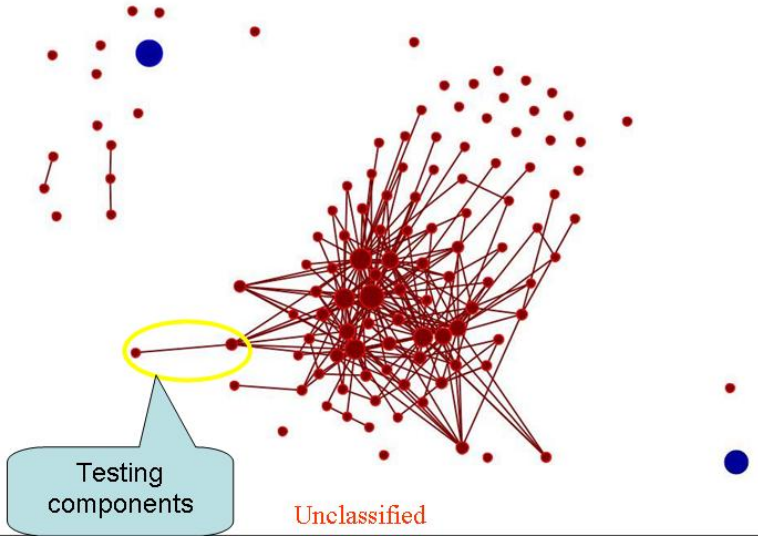
Unclassified

Unclassified

Introduction
The Proposed Approach
Compute the consistency
Identify the properties
Application to the system integration
Conclusion

Part 2: Identify the properties of the elements in the model (Cont'd)

Analysis Via Visualisation (4) - Release 2



Unclassified

Unclassified

Introduction
The Proposed Approach
Compute the consistency
Identify the properties
Application to the system integration
Conclusion

Application of the approach to the system integration

- The major problems involved in system Integration
  - Difficult in modification and maintenance
    - Difficulty in understanding the existing system and interfaces
  - Not coherent and not unifying data structure
  - Many different application and systems to supports
    - incompatible system and/or system interfaces
  - Aging of the systems
  - Social issues
  - ……

Unclassified

Unclassified

Introduction
The Proposed Approach
Compute the consistency
Identify the properties
Application to the system integration
Conclusion

Application of the approach to the system integration (Cont'd)

- The proposed approach is still applicable
  - For integrating the systems with well-defined system models
    - The approach enforces the consistency checking principles
    - Network analysis approach can provide good information about which components (nodes) are vulnerable and with higher complex (higher values of centrality and/or centrality betweens)
  - For integrating a new system to an old system without the well-developed models
    - If the old system can be reverse-engineered
      - some models can be obtained and can be used for analysis as discussed before
    - If it can not be reverse-engineered
      - the old system has to be understood, and architecture level node connections will need to be developed.

Unclassified

Unclassified

Introduction
The Proposed Approach
Compute the consistency
Identify the properties
Application to the system integration
Conclusion

Conclusion

- **Conclusion**:
  - MBSE provides a practical approach to develop complex systems
  - Models produced in the system engineering processes have to be evaluated or assessed to ensure that the requirements are fully implemented, and models are consistent throughout the entire engineering process.
    - The proposed approach is the first step toward addressing the issue
    - More research is required to address other burning issues
  - In order to have better understanding of the system, models have to be studied from holistic level
    - Networks science provides good tools for studying the holistic view of the system, the interconnections, and their changes/evolutions

Unclassified